Concept learning in description logics using refinement operators

DL-Learner: Learning Concepts in Description Logic

TECHNISCHE UNIVERSITÄT DARMSTADT

Inhaltsangabe

TECHNISCHE UNIVERSITÄT DARMSTADT

- Concept learning in description logics using refinement operators
  - Einführung in DL
  - Refinement Operator
  - Lernalgorithmus
  - Evaluation
  - Fazit
- DL-Learner: Learning Concepts in Description Logics
  - Einführung
  - Struktur
  - Fazit

Einführung
Description Logic

TECHNISCHE UNIVERSITÄT DARMSTADT

- Sprache für Wissensrepräsentation
- Untermenge von Prädikatenlogik erster Stufe
  - Weniger Ausdrucksstärke
  - Syntax ohne Variablen
  - Entscheidbar
- Web Ontology Language (OWL) basiert auf DL

Einführung
Description Logic

TECHNISCHE UNIVERSITÄT DARMSTADT

- Konzept
  - Axiom
  - {Mensch, Raum}
- Objekt
  - Konstante
  - gehört einem oder mehreren Konzepten an
  - {E302}
- Rollen
  - Relation

Gustave Courbet -
Selbstportrait mit einem schwarzen Hund

Description Logic
Syntax

TECHNISCHE UNIVERSITÄT DARMSTADT

Syntax	Semantik
A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
$\top$	$\Delta^{\mathcal{I}}$
$\bot$	$\emptyset$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\neg C$	$\neg C = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{a \mid \exists b.(a,b) \in r^{\mathcal{I}} \text{ und } b \in C^{\mathcal{I}}\}$
$\forall r.C$	$(\forall r.C)^{\mathcal{I}} = \{a \mid \forall b.(a,b) \in r^{\mathcal{I}}\}$

Description Logic
Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

Was war auf dem Bild von vor zwei Folien?

Description Logic
Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\text{Potrait} \sqcap \exists \text{beinhaltet.Mann} \sqcap \exists \text{beinhaltet.Hund}$$

Description Logic
Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

Potrait ⊓
∃beinhaltet.Mann ⊓
∃beinhaltet.Hund

Harmensz van Rijn Rembrandt -
Selbstbildnis in orientalischem Kostüm

Jan Wildens - Winterlandschaft mit einem Jäger

Description Logic
Wissen

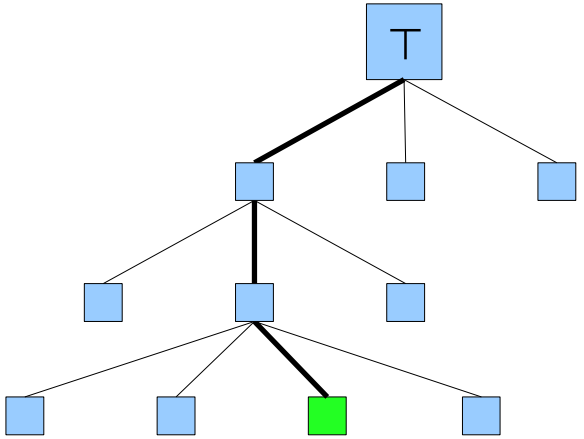
TECHNISCHE UNIVERSITÄT DARMSTADT

- TBox (terminological box)
  - Menge von Konzepten
  - Tisch $\equiv$ Fläche $\sqcap$ $\exists$besitzt.Bein
- ABox (assertional box)
  - Zuordnung Objekte zu Konzepten
  - studiert(Max Mustermann, Informatik)
  - studiertAn(Max Mustermann, TUD)

Refinement Operator
Konzept

TECHNISCHE UNIVERSITÄT DARMSTADT

- Suche in einer Baumstruktur
- Spezialisierung der Aussage
- Ausgangslage: $\top$



Beispiel:

$$\top \rightsquigarrow_\rho \mathrm{Mensch} \rightsquigarrow_\rho \mathrm{Mensch} \sqcap \exists \mathrm{besitzt}.\top \rightsquigarrow_\rho \mathrm{Mensch} \sqcap \exists \mathrm{besitzt}.\mathrm{Hund}$$

Refinement Operator
Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

$\rho(C)$ Menge von Operationen:

$\{\bot\} \cup \rho_T(C)$	falls $C = \top$
$\rho_T(C)$	sonst

$\rho_B(C)$ Menge von Operationen:

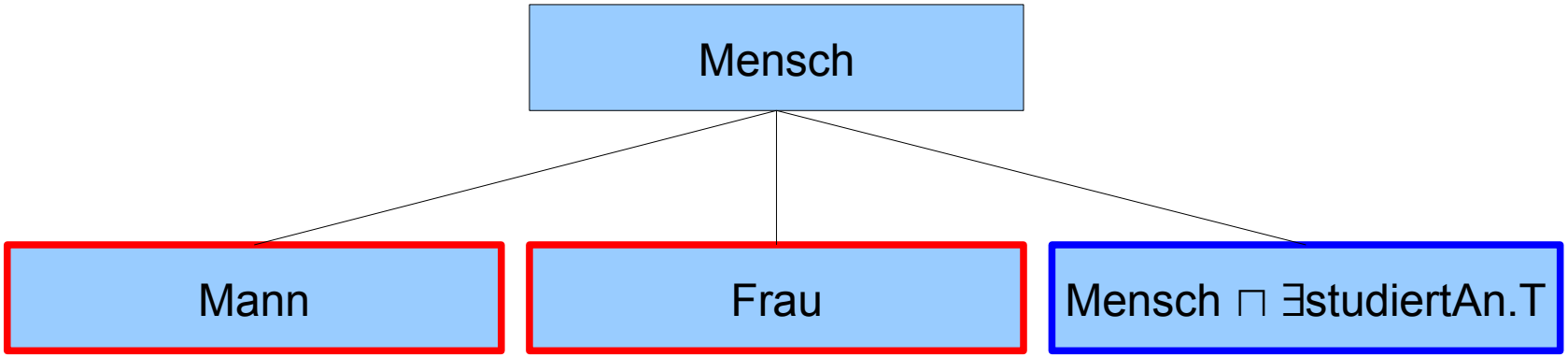
- $\varnothing$ falls $C = \bot$
- $\{C_1 \sqcup \ldots \sqcup C_n \mid C_i \in M_B\ (1 \leq i \leq n)\}$ falls $C = \top$

Refinement Operator
Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\{A' \mid A' \in \mathrm{sh}_{\downarrow}(A)\} \cup \{A \sqcap D \mid D \in \rho_T(\top)\}$$

falls $C = A$ $(A \in N_C)$

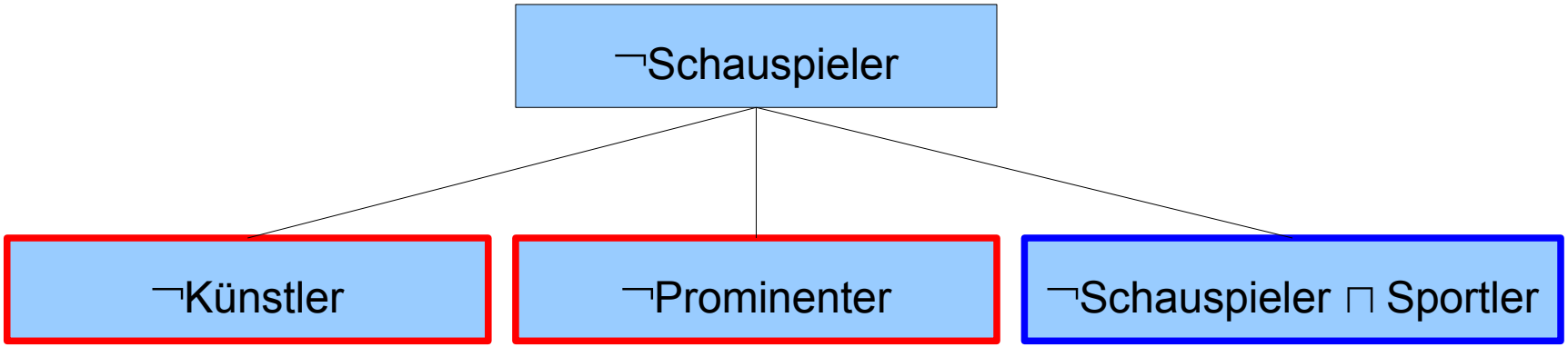


Refinement Operator
Definition (Fortsetzung)

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\{\neg A' \mid A' \in \mathrm{sh}_{\uparrow}(A)\} \cup \{\neg A \sqcap D \mid D \in \rho_T(\top)\}$$

falls $C = \neg A$ $(A \in N_C)$

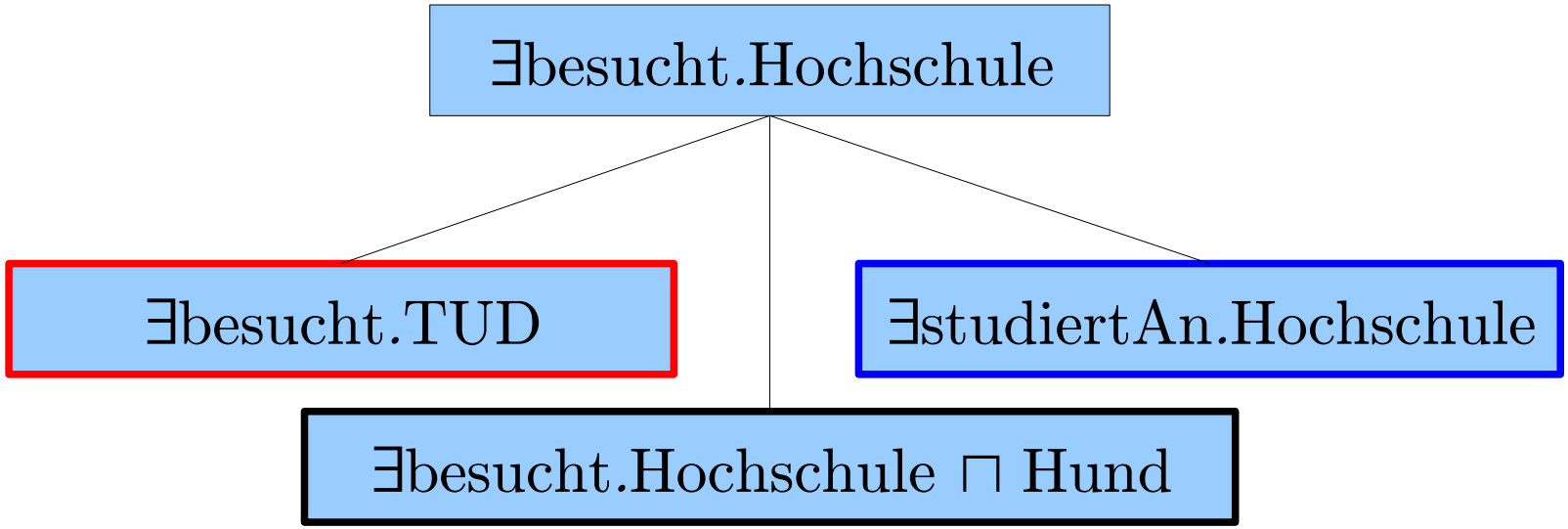


Refinement Operator
Definition (Fortsetzung)

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\{\ \exists r.E \mid A = ar(r),\ E \in \rho_A(D)\}$$
$$\cup \{\exists r.D \sqcap E \mid E \in \rho_B(\top)\}$$
$$\cup \{\exists s.D \mid s \in sh_{\downarrow}(r)\}$$

falls $C = \exists r.D$



Refinement Operator
Definition (Fortsetzung)

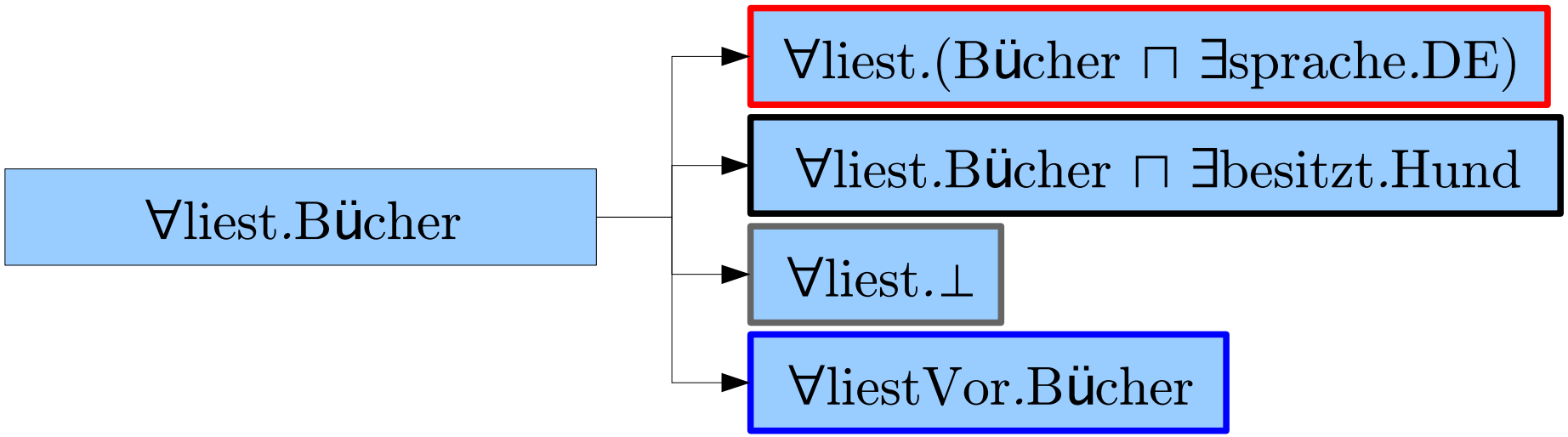
TECHNISCHE UNIVERSITÄT DARMSTADT

$$\{\ \forall r.E \mid A = ar(r),\ E \in \rho_A(D)\}$$ falls $C = \forall r.D$

$$\cup \{\forall r.D \sqcap E \mid E \in \rho_B(\top)\}$$

$$\cup \{\forall r.\bot \mid D = A \in N_C \text{ und } sh_{\downarrow}(A) = \varnothing\}$$

$$\cup \{\forall s.D \mid s \in sh_{\downarrow}(r)\}$$



Refinement Operator
Definition (Fortsetzung)

TECHNISCHE UNIVERSITÄT DARMSTADT

- $\{C_1 \sqcap \ldots \sqcap C_{i-1} \sqcap D \sqcap C_{i+1} \sqcap \ldots \sqcap C_n \mid D \in \rho_B(C_i),\ 1 \leq i \leq n\}$

  falls $C_1 \sqcap \ldots \sqcap C_n (\mathrm{n} \geq 2)$

- $\{C_1 \sqcup \ldots \sqcup C_{i-1} \sqcup D \sqcup C_{i+1} \sqcup \ldots \sqcup C_n \mid D \in \rho_B(C_i),\ 1 \leq i \leq n\}$
  $\cup\{(C_1 \sqcup \ldots \sqcup C_n) \sqcap D \mid D \in \rho_B(\top)\}$

  falls $C_1 \sqcup \ldots \sqcup C_n (\mathrm{n} \geq 2)$

Refinement Operator
Probleme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Unendlich
  - Bsp.: $\forall r.\forall r.\ \ldots\ C$
  - Lösung: Beschränkung auf Länge n
- Redundant
  - Mehrere Pfade im Suchbaum zu gleichem Konzept
  - Lösung: ein Redundanzcheck
- Nicht zweckmäßig (not proper)
  - $C \rightsquigarrow_\rho C_1 \rightsquigarrow_\rho \ldots \rightsquigarrow_\rho C_n = D$

    $mit\ C \not\equiv D\ und\ C_j \equiv C\ für\ 1 \leq i \leq n\text{-}1$
  - Lösung: $\rho^{cl}_{\downarrow}(C)$

Lernalgorithmus
Heuristik

TECHNISCHE UNIVERSITÄT DARMSTADT

- accurancy(C) = 1 – (up + cn)/|E|
- acc_gain(N) = accurancy(C) – accurancy(C')
  - C' ist das Konzept des Elternknoten
- Bewertung(N) = accuracy(C) + α*acc_gain(N) + β*n

Lernalgorithmus
Implementierung

TECHNISCHE UNIVERSITÄT DARMSTADT

Input: Wissensdatenbank, Beispiele

SB (Suchbaum) bestehend nur aus dem Wurzelknoten ($\top$, 0, *false*)

while *SB* beinhaltet keinen passenden Knoten **do**

wähle N = (C, n, b) mit höchster Bewertung in *SB*

erweitere die Länge von N auf n + 1:

begin

füge alle (D, n, checkRed(SB,D)) mit

$D \in \text{trans}(\rho^{cl}_{\downarrow}(C))$ and $|D| = n + 1$ als Kinder N hinzu

überprüfe die neuen Knoten auf Redundanz

ändere N zu (C, n + 1, b)

end

Return passende Konzept in SB

Evaluierung
Aufbau

TECHNISCHE UNIVERSITÄT DARMSTADT

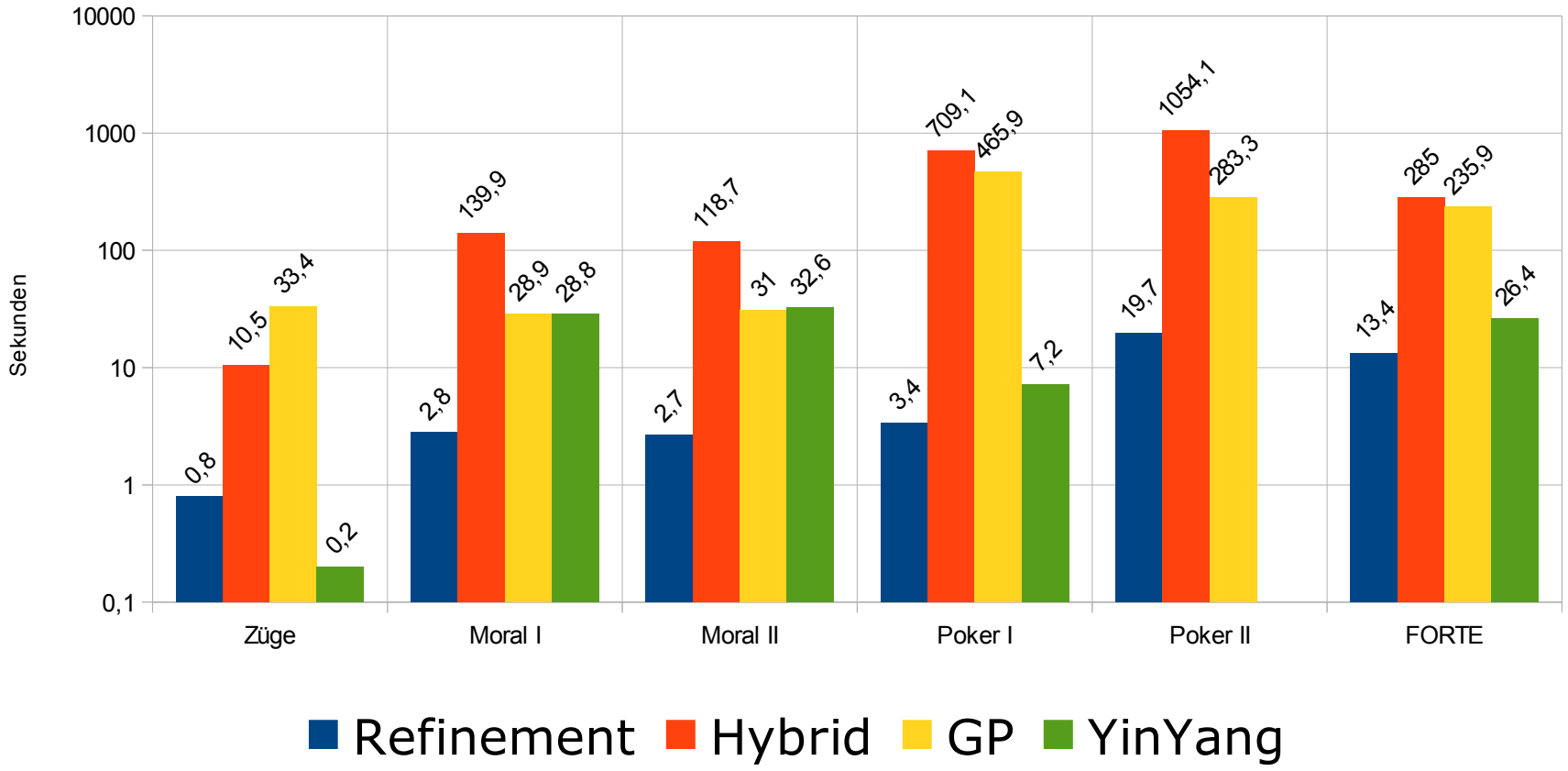
- Fehlen an Trainingsdaten für Ontologielernen
  - Konvertierung existierender Lernprobleme in OWL
- Michalski's Trains [2]
  - Züge mit unterschiedlichen Wagons
- FORTE [1]
  - Familienbeziehungen
- UCI Machine Lerning repository
  - Hand beim Poker
  - Moralische Argumentation von Menschen
  - http://archive.ics.uci.edu/ml/

Evaluierung
Vergleich

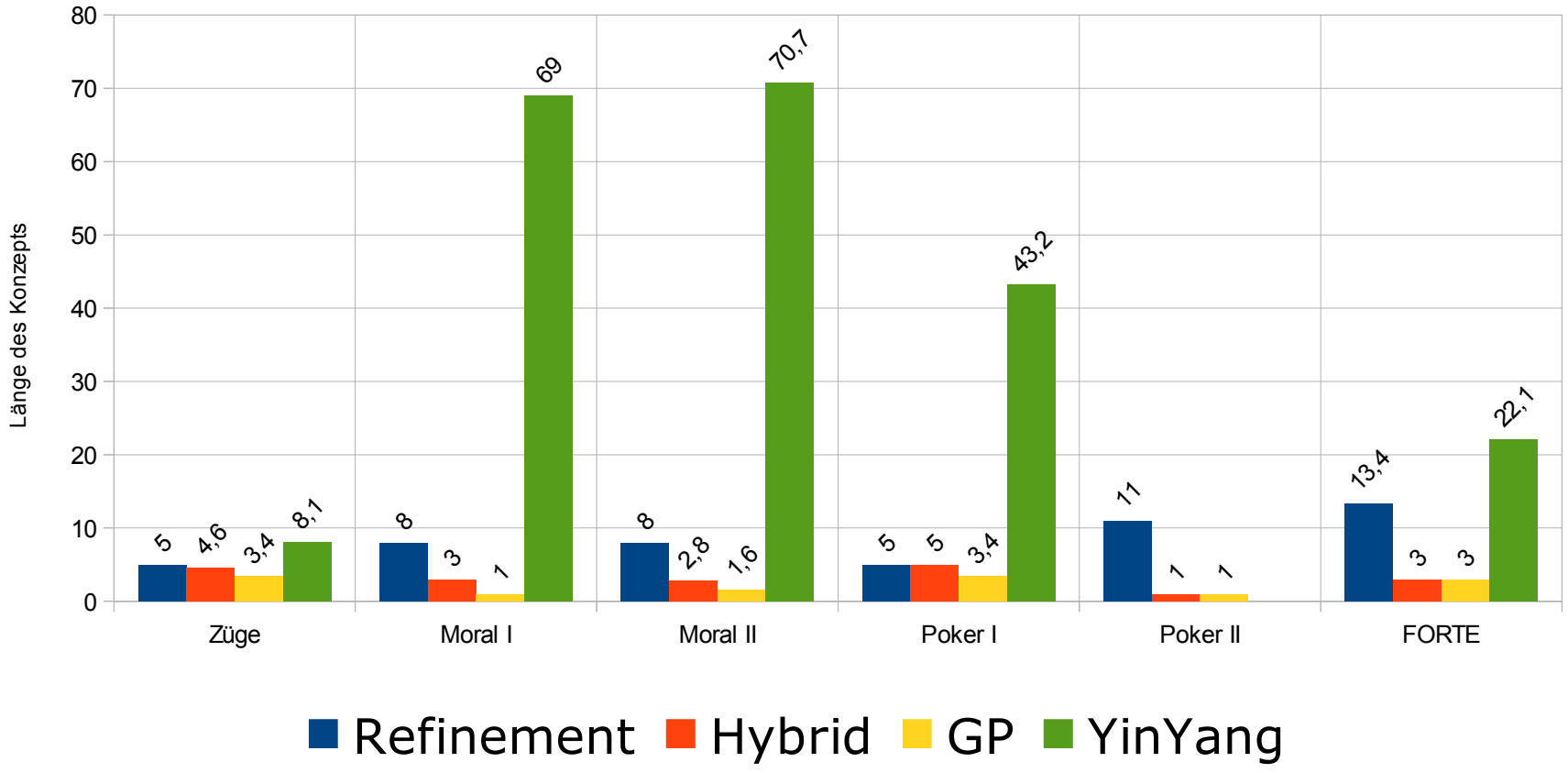
TECHNISCHE UNIVERSITÄT DARMSTADT

- YinYang [3]
- Hybrider Algorithmus
  - Refinement Operator + generische Programmierung
- Generische Programmierung

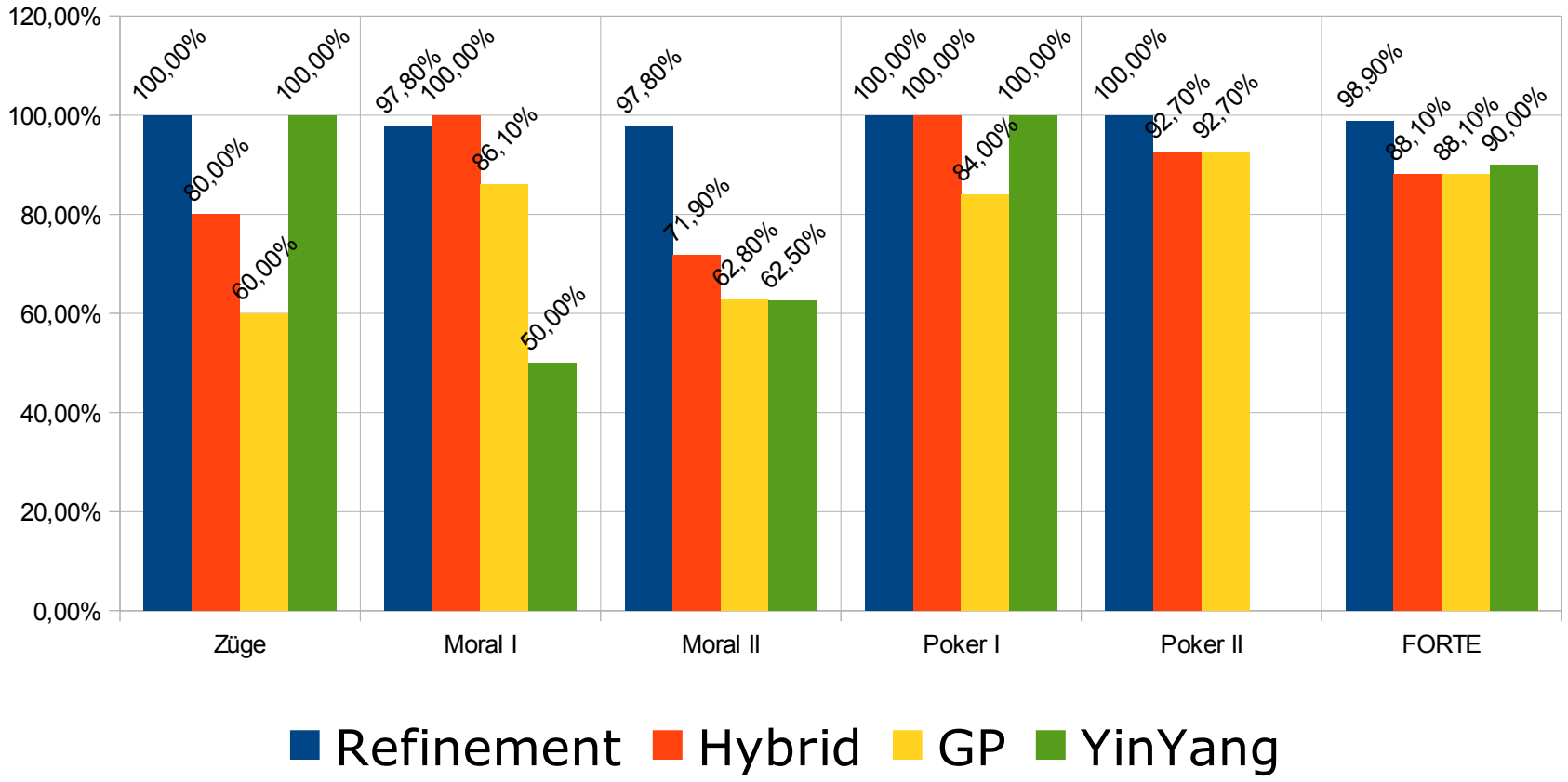
Evaluierung
Dauer



Evaluierung
Länge



Evaluierung
Accuracy



Kritik

TECHNISCHE UNIVERSITÄT DARMSTADT

- Unbefriedigende Evaluierungsszenarien
- Nicht bewiesener Redundanzcheck

Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

- Erstellung von kompakten lesbaren Konzepten
- Hohe Accuracy
- Anwendbar auf beliebige Datensätze
  - Bsp.: DBpedia mit 100 Millionen Axiome

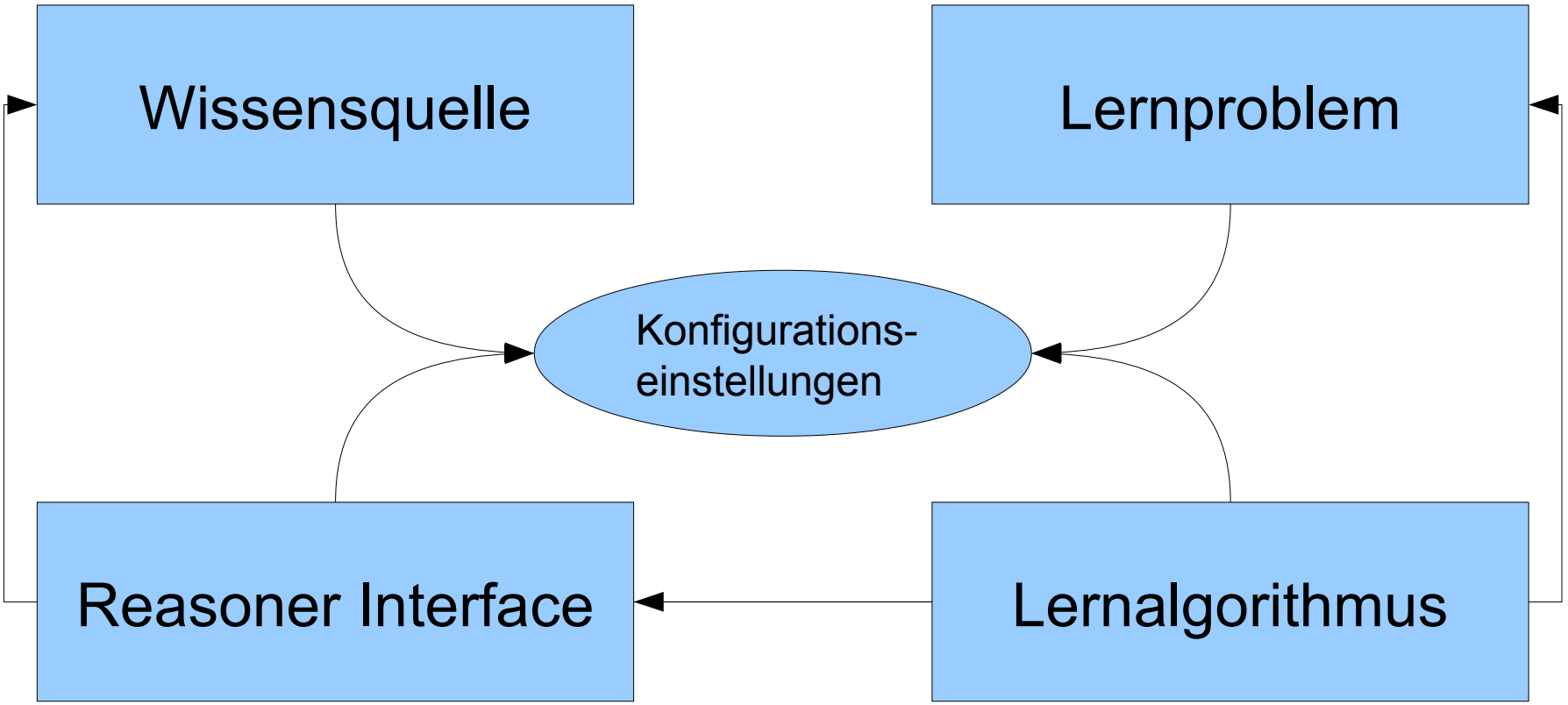
DL-Learner
Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Framework für Lernalgorithmen
- Spezialisiert auf OWL
- Open Source
- Java 6
- http://dl-learner.org/Projects/DLLearner

DL-Learner
Struktur

TECHNISCHE UNIVERSITÄT DARMSTADT



Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gut dokumentiert
  - Inklusive einer Wiki
- Modularer Aufbau
  - Leicht erweiterbar

Ende

TECHNISCHE UNIVERSITÄT DARMSTADT

Fragen?

Anhang

Literatur

[1] Richards, B. L., & Mooney, R. J. (1995). Refinement of first-order Horn-clause domain theories. Machine Learning, 19(2), 95–131.

[2] Michalski, R. S. (1980). Pattern recognition as rule-guided inductive inference. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2(4), 349–361.

[3] Iannone, L., Palmisano, I., & Fanizzi, N. (2007). An algorithm based on counterfactuals for concept learning in the semantic web. Applied Intelligence, 26(2), 139–159.

Refinement Operator
Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

$\rho(C)$ Menge von Operationen:

$\{\bot\} \cup \rho_T(C)$	falls $C = \top$
$\rho_T(C)$	sonst

$\rho_B(C)$ Menge von Operationen:

- $\varnothing$ falls $C = \bot$
- $\{C_1 \sqcup \ldots \sqcup C_n \mid C_i \in M_B\ (1 \leq i \leq n)\}$ falls $C = \top$
- $\{A' \mid A' \in \mathrm{sh}_{\downarrow}(A)\} \cup \{A \sqcap D \mid D \in \rho_T(\top)\}$ falls $C = A\ (A \in N_C)$

Refinement Operator
Definition (Fortsetzung)

TECHNISCHE UNIVERSITÄT DARMSTADT

- $\{\neg A' \mid A' \in \mathrm{sh}_{\uparrow}(A)\}$ falls $C = \neg A$ $(A \in N_C)$

  $\cup \{\neg A \sqcap D \mid D \in \rho_{\top}(\top)\}$
- $\{\ \exists r.E \mid A = ar(r),\ E \in \rho_A(D)\}$ falls $C = \exists r.D$

  $\cup \{\exists r.D \sqcap E \mid E \in \rho_B(\top)\}$

  $\cup \{\exists s.D \mid s \in sh_{\downarrow}(r)\}$
- $\{\ \forall r.E \mid A = ar(r),\ E \in \rho_A(D)\}$ falls $C = \forall r.D$

  $\cup \{\forall r.D \sqcap E \mid E \in \rho_B(\top)\}$

  $\cup \{\forall r.\bot \mid D = A \in N_C \text{ und } sh_{\downarrow}(A) = \varnothing\}$

  $\cup \{\forall s.D \mid s \in sh_{\downarrow}(r)\}$

Refinement Operator
Definition (Fortsetzung)

TECHNISCHE UNIVERSITÄT DARMSTADT

- $\{ C_1 \sqcap \ldots \sqcap C_{i-1} \sqcap D \sqcap C_{i+1} \sqcap \ldots \sqcap C_n \mid D \in \rho_B(C_i),\ 1 \leq i \leq n \}$

  falls $C_1 \sqcap \ldots \sqcap C_n (\mathrm{n} \geq 2)$
- $\{ C_1 \sqcup \ldots \sqcup C_{i-1} \sqcup D \sqcup C_{i+1} \sqcup \ldots \sqcup C_n \mid D \in \rho_B(C_i),\ 1 \leq i \leq n \}$
  $\cup \{ (C_1 \sqcup \ldots \sqcup C_n) \sqcap D \mid D \in \rho_B(\top) \}$

  falls $C_1 \sqcup \ldots \sqcup C_n (\mathrm{n} \geq 2)$